



Detecting Mobile Crowdsensing Context in the Wild

Rachit Agarwal, Shaan Chopra, Vassilis Christophides, Nikolaos Georgantas,
Valérie Issarny

► To cite this version:

Rachit Agarwal, Shaan Chopra, Vassilis Christophides, Nikolaos Georgantas, Valérie Issarny. Detecting Mobile Crowdsensing Context in the Wild. 20th IEEE International Conference on Mobile Data Management (MDM) 2019, Jun 2019, Hong Kong, Hong Kong SAR China. hal-02151434

HAL Id: hal-02151434

<https://inria.hal.science/hal-02151434>

Submitted on 11 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Detecting Mobile Crowdsensing Context in the Wild

Rachit Agarwal, Shaan Chopra, Vassilis Christophides, Nikolaos Georgantas, Valérie Issarny

Inria-Paris, France

Email: {rachit.agarwal, shaan.chopra, vassilis.christophides, nikolaos.georgantas, valerie.issarny}@inria.fr

Abstract—Understanding the sensing context of raw data is crucial for assessing the quality of large crowdsourced spatio-temporal datasets and supporting context-augmented personal trajectories. Detecting sensing contexts in the wild is a challenging task and requires features from smartphone sensors that are not always available. In this paper, we propose three heuristic algorithms for detecting sensing contexts such as in/out-pocket, under/over-ground, and in/out-door for crowdsourced spatio-temporal datasets. These are unsupervised binary classifiers with a small memory footprint and execution time. Using a segment of the Ambiciti real dataset – a feature-limited crowdsourced dataset – we report that our algorithms perform equally well in terms of balanced accuracy (within 4.3%) when compared to machine learning (ML) models reported by an AutoML tool.

Index Terms—Context, Urban Sensing

I. INTRODUCTION

With the rapid growth of smart devices, equipped with sensing, computation, and communication capabilities, *mobile crowdsensing* [1] emerges as an important paradigm to acquire high-resolution spatio-temporal data from physical environments (e.g., traffic conditions, urban air, and noise pollution, etc.) at a low cost. For example, the microphone of smartphones can be used as an acoustic sensor [2], while location information of such crowd-measurements can be easily obtained using on-board GPS receivers, remote GPS location servers or nearby WiFi networks. However, despite recent advances in positioning techniques [3], location accuracy obtained by today's smartphones on the market is constrained by several behavioral factors related to *how* users employ smart devices in the wild: (a) whether the user was located *in* vs. *out* door; or (b) was moving *under* vs. *over* ground. Accuracy problems may also be encountered in noise measurements if the smartphone is kept *in* vs. *out* of the pocket.

As a matter of fact, human behavior factors (e.g., holding position of the smartphone, human movement during sensing, etc.) are increasingly recognized as important causes of unreliable crowd-measurements besides power failures, intermittent communications, or sensor calibration [4]. Hence, understanding the sensing context of raw crowd-measurements is crucial in managing the quality of large spatio-temporal datasets collected via crowdsensing [5] but also in enriching personal trajectories with transportation modalities and speed [6] and support context-adaptive virtual assistants [7]. Detecting sensing context in-the-wild is a challenging task as it requires to map a large variety of features extracted from incomplete and unbalanced sensor data to a number of predicted context labels [8] using very limited ground truth.

In this paper, we focus on lightweight methods for detecting the sensing context of crowd measurements obtained

via commonly used on-board sensors of smartphones (e.g., proximity, GPS). More precisely, we are interested in (a) the *in/out-pocket* context as it may enhance the precision of inferring user's mobility activity¹ [9], (b) the *under/over-ground* context as it may inform location-based services for the cause of the very bad GPS accuracy e.g., in metro stations [10], and (c) the *in/out-door* context as it enables to better understand distance information². Most of the publicly available crowdsensing datasets (e.g. PRIVA'MOV [11] and Beijing taxi dataset [12]) do not include data from specialized sensors such as light, barometer, etc. considered by state-of-the-art context detection algorithms (see also Table I). As contextual information is needed to assess the reliability of crowd-measurements consumed by mobility applications, we are interested in mining the sensing context using feature-limited crowdsensing datasets commonly available.

In particular, we are proposing heuristic-based binary classification algorithms for inferring the three sensing contexts mentioned above. We particularly pay attention to minimizing the tuning efforts of the algorithms and thus be able to deploy them on users' smartphones easily. As the lack of certain features prohibits us to consider state-of-the-art algorithms as baselines, we compare the performance of our heuristic algorithms against Machine Learning (ML) models built using the same set of features by an AutoML tool [13].

In a nutshell, we make the following contributions:

- We propose three **unsupervised binary classifiers** for annotating crowdsourced datasets with **contextual information** (i.e., in/out-pocket, under/over-ground and in/out-door) **using a limited set of features** that are commonly available in these datasets.
- Our experimental evaluation with a segment of the Ambiciti real dataset demonstrates that the in/out-pocket, under/over-ground and in/out-door contexts can be detected with a balanced accuracy of around 54%, 70.2% and 65%, respectively. When compared with the best baseline ML model in terms of balanced accuracy, the heuristic algorithms for in/out-pocket perform equally well, while for under/over-ground and in/out-door contexts for a distance threshold value $\tau^{uo}=313\text{m}$ it achieves 4.3% and 1%, respectively, lower.

The rest of the paper is organized as follows. Section II details the set of features and achieved performance of state-

¹Accelerometer which is mainly used for detecting the mobility activity can report differently when phone is in-pocket and when it is held in hand

²when indoor, jump-lengths are short and have high frequency

of-the-art methods for mining sensing context. Section III describes our heuristic algorithms while Section IV reports the results of our experimental evaluation using a segment of the Ambiciti real dataset annotated with ground truth provided by volunteer users. Section V summarizes our results and presents directions for future work.

II. RELATED WORK

Several algorithms have been proposed for identifying in/out-pocket, under/over-ground, and in/out-door contexts.

1) *In/Out-Pocket context detection*: Yang et al. [14] perform conditional checks on the light intensity and proximity distance obtained using the ambient light and proximity sensors of a mobile device to identify in/out-pocket context. On the other hand, Kawahara et al. [15] infer the position of a mobile device on a user's body (e.g., in chest pocket, in a pant pocket, in a bag, in hand) from the variance/change in the angle formed by the mobile device with the gravity vector. For example, when the mobile device is kept in the trousers pocket, the angle formed fluctuates a lot when the user is walking. In a different way, Miluzzo et al. [9] exploit the data collected using the microphone sensor to infer the in/out-pocket context. They extract multiple features from the recorded audio clip to infer context in a cost-efficient manner via classification.

2) *Under/over-ground context detection*: Erum et al. [10] exploit the air pressure "trapped" between the walls of a narrow tunnel as a train moves underground, also known as the piston effect. The authors experiment with two types of pressure sensors (Bosch BMP180 available in Google Nexus 4 and BMP280 available in iPhone 6) to identify context related to underground stations and trips in the London and Brussels metro.

3) *In/out-door detection*: IODetector [16] is a sensing service which builds upon the fact that light, LTE received signal strength (RSSI) and magnetic field exhibit different behaviors in in/out-door settings. IODetector uses Hidden Markov Model (HMM) to identify the probability of being inside or outside. MobiIO [17], on the other hand, is motivated by the following assumptions: (1) when users are outside, they usually do not sit, and (2) users tend to make relatively more turns when they are inside rather than when outside. In a different way, Okamoto et al. [18] use the S/N ratio and altitude information captured by the GPS sensor to study the detection of in/out-door. They further improve the detection accuracy by using the information captured by the compass sensor. Wang et al. [19] detect in/out-door by classifying the RSSI of neighboring LTE base stations in varying environments. They identify that K-Nearest Neighbors (KNN) achieves the best accuracy among other ML models and detects in/out-door within 8 sec. Primarily aiming to improve the energy consumption of GPS based methods, Chen et al. [20] modified the GPS algorithm to include the dynamic grouping of satellites and propose SatProbe which also detects in/out-door. They show that their algorithm requires only 0.35 sec to detect in/out-door. Li et al. [21] propose an aggregated IOS (indoor, open space and near buildings)

TABLE I
FEATURES USED IN RELATED STUDIES

Features	Sensor used	In/out-door	Under/over-ground	In/out-pocket
Light intensity	Light	[16], [22] [21], [23]	-	[14]
Magnetic Strength	Magnetometer	[16], [22]	-	-
WiFi RSSI	WiFi*	[21], [23]	-	-
Proximity distance	Proximity	[16]	-	[14]
RSSI level	GPS/GSM	[16], [19] [20], [22], [23]	-	-
Time of the day	Clock*	[16]	-	-
Pressure	Barometer	-	[10]	-
Noise level	Microphone	-	-	[9]
Mobility Activity	Accelerometer	[17], [23]	-	-
Acceleration	Accelerometer	[16], [17]	-	[15]
Altitude	GPS	[18], [20]	-	-
S/N Ratio	GPS†	[18], [20]	-	-
Direction	Compass	[18]	-	-
# turns when moving	O-Sensor ⁺	[17]	-	-

+: Direction sensor reported in [17], *: Sensor used is not mentioned in the related study, †: GPS sensor captured satellite SNR

detection. They use semi-Markov conditional random fields (CRF) with light intensity and WiFi RSSI and show that it achieves significant improvement over an Adaptive Boosting (AdaBoost) learner with WiFi RSSI. In a similar study, Liu et al. [22] infer in/out-door context using light intensity and proximity. To deal with cases where the device is kept inside a pocket or bag, they enrich the feature space with the LTE cell identity map (CIMAP), which comprises cell identity, cell type, and RSSI. Using a similar approach, Ali et al. [23] present SenseIO that maps single serving LTE cell tower, WiFi, light density, and recognized activity state to identify in/out-door contexts. They experimentally demonstrate that SenseIO is capable of inferring fine-grained contexts (rural, urban, indoor, and complex places).

We summarize in Table I the rich feature space that custom algorithms rely upon for evaluation. These algorithms thus employ custom datasets that are carefully crafted in controlled settings (see Table II). In table II we also summarize the techniques used by these algorithms.

III. CONTEXT MINING

In most of the existing datasets many of the features listed in Table I are not available. Therefore, to identify in/out-pocket, under/over-ground and in/out-door contexts we mainly focus on more commonly available features. Next, we present our heuristic algorithms for the three contexts.

1) *In/out-Pocket context detection*: Proximity sensing can serve as a feature for determining the in/out-pocket context. If the crowdsourcing device, say a smartphone, is in-proximity of some object (can be inside the pocket, bag or phone cover), it can be labeled as in-pocket. But certain corner cases do exist. For example, if the user is holding the phone too close to their ear while talking, the proximity sensor is active. This

TABLE II
DATA COLLECTED AND ALGORITHM USED BY RELATED STUDIES

Related study	Uses	Data collection			Reported accuracy in %	Power, energy, Battery duration, CPU consumption
		# of devices/users	Duration	Other details		
[14]	Conditional checks	no training needed			>98	6mW, CPU≈0
[15]	Variance FFT Angle	4	10m	-	>96	-
[9]	Temporal smoothing GMM, SVM	2	14m	‡	≈80	-
[10]	moving avg.	2	3d	20+19 ⁺	>55.6 [†]	-
[16]	HMM	3	30d	19 ⁺	>82	>7.2hr
[17]	HMM, SVM, sliding window	-	1d	-	≈75	5.7mAh
[18]	SVM	19	169m	-	>86* >96	-
[19]	KNN	1	-	-	97.27	-
[20]	modified GPS info. detection	1	-	79 ⁺ , 2595 ^p	85.6	143.1mW
[21]	semi-Markov CRF	3	env. dependent		96	>13.7hr
[22]	CIMAP	1	-	-	>98	4mW
[23]	conditional checks	no training needed			>92	-

d: days, m: minutes, s: seconds, *: baseline, +: trajectories, p: points in diverse environment, ‡: 116k samples (50% used in training), †: for different cities and cases different accuracies are reported.

causes the collected data points to be mislabelled. However, depending on the presence of other parameters, the algorithm can be modified. Say, the measurement mode that reflects how measurements are recorded (e.g., ‘manually’ or ‘automatically’) is known. Then, a manually recorded measurement can be labeled as out-pocket (a user is required to tap onto the screen to take the measurement which is more conveniently done when the smartphone is held in hand). We design our

```

inPocket = False;
if measurement is made "manually" then
    inPocket = False;
else
    if proximity == True then
        inPocket = True;
    else
        inPocket = False;
    end
end
end

```

Algorithm 1: In/out-pocket assignment

heuristic algorithm (see Algorithm 1) to take into account the above aspects. If the *measurement is made ‘manually’*, the algorithm labels the data point as out-pocket. If not, the algorithm checks the value of the *proximity* attribute; if the phone is in-proximity of some object (can be inside the pocket or bag or cover), it is labeled as in-pocket, else out-pocket.

2) *Under/over-ground context detection:* For determining the under/over-ground context, altitude at which the measurement is taken is an important factor. However, altitude, if

available, along with latitude-longitude information, may not be accurate and sufficient to determine whether the measurement is taken under-ground or over-ground. GPS identified altitude measurements are not always accurate. To tackle such problem, other assumptions and inferences are needed. For example, most of the metro stations are underground. Data points within a certain distance from such stations can be assumed to be at the same altitude. However, corner cases exist such as when the person is in a building above an underground station, outside but near an underground station, and in an underground road. We, thus, design our heuristics

```

underground = False;
if altitude > 0 then
    underground = False;
else if altitude < 0 then
    underground = True;
else
    # also valid in cases when altitude is not given;
    if point near an underground metro station then
        underground = True;
    else
        underground = False;
    end
end
end

```

Algorithm 2: Under/Over-ground assignment

algorithm (see Algorithm 2) to take into account above aspects. If the altitude<0, the context is marked as underground. If altitude>0, it is marked as over-ground. If altitude=0 or if it is not provided, the algorithm uses the distance from a metro station as a factor. If the measurement is taken at a point that lies within a defined threshold, the point is considered to be at the same level as that metro station otherwise it is considered as over-ground.

```

indoor = False;
if underground == True then
    indoor = True;
else
    if user recognized activity is still or stationary then
        indoor = True;
    else
        if connected via WiFi then
            indoor = True;
        else
            indoor = False;
        end
    end
end
end

```

Algorithm 3: In/Out-door assignment

3) *In/out-door detection:* For determining the in/out-door context, along with GPS coordinates, altitude is a decisive factor as we can consider all underground measurements as indoor. Moreover, the availability of parameters such as speed and recognized activity (still, walking, in a vehicle such as a car, train or metro) of the user, can be used to enhance the detection. Note that currently as an in-built API the mobile device OS already provides recognized activity information. Another feature that affects this context is the network (WiFi or mobile data) using which the measurement is taken. If

available, such features should ideally enhance the context detection performance.

Thus, we base our heuristics algorithm (see Algorithm 3) on these features. Firstly, if the measurement is underground, it is considered indoor. If not, then the algorithm checks for *users recognized activity*. If it is provided and is *still* or *stationary*, the measurement is labeled indoor. Otherwise, the algorithm checks for the attribute that provides what type of network is used to capture the measurement; if it is *WiFi*, then the point is labeled indoor, else as outdoor. Though, we acknowledge that there are multiple assumptions involved in this approach and corner cases, such as people walking/traveling indoor and *WiFi* present outdoor, can be mislabeled.

IV. EVALUATION

We evaluate the effectiveness of our heuristics-based algorithms on a real dataset gathered using Ambiciti³ application. We ask users of Ambiciti to volunteer and, additionally, manually annotate data points with sensing context (provide ground truth data) as the data points are collected. We evaluate the performance of our detection algorithms by identifying the *accuracy*, *precision*, *recall*, and *F1* scores. We additionally compare the performance of our algorithms against ML models built using the same set of features for each context. To construct these baselines models, we rely on an AutoML tool called Tree-Based Pipeline Optimization Tool (TPOT) [13] that selects the ML algorithm that achieves the best classification accuracy for a particular dataset. For our performance evaluation, we train the ML algorithms on 80% of the data points. Further, we tune TPOT on its default parameters. We implement all the algorithms in Python.

A. Dataset

The ground truth is of the form $O=\{ID, \vec{l}, t, \vec{v}\}$, where ID is the anonymized deviceID, \vec{l} is the vector containing GPS coordinates ($\vec{l}=\{latitude, longitude, altitude\}$), t is the time of measurement, and \vec{v} is the vector that contains other attributes along with ground truth attributes ($\vec{v}=\{proximity, measurement_mode, in-pocket, in-door, under-ground, recognized_activity, connected_via\}$). Here, latitude and longitude are in decimal degrees. Proximity, in-pocket, indoor and underground are binary. For example a `True` value for in-pocket represents that the data point was collected when the device was inside the pocket. The in-pocket, indoor and underground attributes holds the ground truth values that the volunteer users provided. `Measurement_mode` \in [`manually`, `automatically`]. `Recognized_activity` \in [`still`, `stationary`, `sitting`, `tilting`, `on foot`, `walking`, `on bike`, `in vehicle`] and `connected_via` \in [`WiFi`, `others`]. For a complete detailed description of the Ambiciti platform and the application, we refer readers to [24].

The ground truth data consists of 20,046 data points collected by 7 volunteers based in Paris. It has a temporal granularity of 5 seconds and is collected over a period of 3 months (from 1st May 2018 until 30th July 2018). In this

dataset, the distribution of data points is not balanced for the different context. Out of 20,046 data points, there are only 4838 in-pocket, 3310 under-ground, and 5672 indoor data points. Such imbalance within contexts motivates us to identify *balanced accuracy* as well.

B. Results

1) *In/out-pocket*: For detecting in/out-pocket, we use *measurement_mode* and *proximity* feature. According to TPOT, using the above features, Gaussian Naive Bayes (NB)⁴ (*priors=None*, *var_smoothing*= 10^{-9}) achieves the best balanced accuracy while Decision tree classifier achieved the best accuracy in the unbalanced case. When comparing the above with the balanced accuracy obtained using our algorithm (see Algorithm 1), we identify that both cases perform equally well (see Table III). We note that Gaussian NB outperforms our proposed heuristic technique in terms of in-pocket precision, recall, and F1 score, while the heuristic algorithm performs better for the out-pocket case (mostly owing to the corner cases discussed previously).

2) *Under/over-ground*: Algorithm 2 requires *altitude* feature. It needs *latitude* and *longitude* to identify distance. It also needs a distance threshold (τ^{uo}) to label points near underground metro stations. As the volunteers are based in Paris, we use the Paris metro stations dataset⁵ that provides under/over-ground tags and center location coordinates for each station but not the depth or altitude at which each station is located. Thus, we do not consider the corner cases described in Section III-2. To identify the distance between a metro station and a point we use Haversine distance method.

Most of the metro stations in the Paris region are $\approx 500m$ apart. We first study the effect of τ^{uo} on balanced accuracy achieved by our algorithm. We identify that the best balanced accuracy is achieved when the distance threshold $\tau^{uo}=313m$ for our algorithm (see figure 1). Next, we use TPOT to identify best ML model at different τ^{uo} 's. For different τ^{uo} 's, TPOT reports, in most of the cases, Bernoulli NB with parameters $\alpha=1$, *binarize*=0.0, *class_prior=None*, *fit_prior=False* from a set of ML models. TPOT also reports Bernoulli NB to produce best accuracy in the unbalanced case but with *fit_prior=True* rest all the parameters are same. At different τ^{uo} , Bernoulli NB, with above-mentioned parameters, provides balanced accuracy within $[-0.22, 8.70]$ of our heuristics method (see figure 1). Here, a negative value indicates that our proposed algorithm performed better than the Bernoulli NB and a positive value indicates that Bernoulli NB performed better. For $\tau^{uo}=313m$ our algorithm identifies the best balanced accuracy which is 4.3% less than what Bernoulli NB achieves. We see that Bernoulli NB outperforms our proposed heuristic technique in terms of overground precision and F1 score, while for the recall the heuristic algorithm

⁴To understand all the hyperparameters obtained we refer readers to <https://scikit-learn.org/stable/>

⁵This dataset is available at <https://github.com/ragarwa2/ile-de-france-Stations>.

³<http://ambiciti.io> - A commercial crowdsourced based noise environment monitoring application

TABLE III
ACCURACY, PRECISION, RECALL AND F1 SCORE REPORTED BY DIFFERENT METHODS.

	Method	τ^{uo} in m	Accuracy in %		Precision		Recall		F1 score	
			balanced	unbalanced	in	out	in	out	in	out
In/Out-Pocket	Gaussian NB [†]	-	54	-	0.33	0.78	0.19	0.89	0.24	0.83
	Heuristics	-	54	-	0.19	0.89	0.19	0.89	0.19	0.89
	DecisionTree [†]	-	-	75.6	0.0	0.77	0.0	1.0	0.0	0.87
Under/Over-ground	Bernoulli NB [†]	313	74.5	-	0.33	0.97	0.81	0.68	0.46	0.79
	Heuristics	313	70.2	-	0.62	0.78	0.62	0.78	0.62	0.78
	Bernoulli NB [†]	313	-	83	0.0	0.83	0.0	1.0	0.0	0.90
In/Out-door	Bernoulli NB [†]	313	66	-	0.42	0.84	0.70	0.62	0.53	0.71
	Heuristics	313	65	-	0.62	0.68	0.62	0.68	0.62	0.68
	Gradient boosting [†]	313	-	71.6	0.0	0.71	0.0	1.0	0.0	0.83

[†] TPOT reported method

outperforms for the overground case (mostly owing to the corner cases discussed previously).

3) *In/out-door*: For this case, we use *recognized_activity*, *connected_via*, and *underground* at $\tau^{uo}=313$ m features. Note that here we use the results of our under/over-ground heuristics algorithm. In this case, similar to the previous one, according to TPOT, Bernoulli NB ($\alpha=0.001$, *binarize*=0.0, *fit_prior*=False, *class_prior*=None) achieves the best balanced accuracy while Gradient Boosting classifier achieves the best unbalanced accuracy. When comparing the best balanced accuracy obtained for all the cases, we see that our heuristics algorithm performs equally well as compared to Bernoulli NB (see Table III). We note that Bernoulli NB outperforms our proposed heuristic technique in terms of out-door precision and F1 score, while for recall, the heuristic algorithm outperforms the out-pocket case.

We further perform execution time and memory consumption analysis for our heuristic algorithms. We use the *memory_profiler*⁶ package of python to obtain memory statistics. Concerning memory, our heuristics algorithms in/out-pocket, under/over-ground and in/out-door require 0kB, 4kB, and 0kB, respectively to detect the context. This memory consumption is w.r.t. only our algorithms and not w.r.t. storage of data points. Our under/over-ground context detection algorithm requires memory as it needs to identify distances of metro stations from the data points. These three algorithms take 0.08sec, 0.17sec and 0.003sec, respectively, for execution.

V. CONCLUSION AND FUTURE WORK

Despite the limited set of features used in heuristics-based classification, we can still mine sensing contexts at an acceptable performance (i.e., above a random binary classifier) compared to state-of-the-art algorithms that use special features such as light, barometer, etc. As demonstrated in our experimental evaluation, using the limited set of features of existing crowdsensing datasets, our heuristic algorithms achieve an equivalent performance for the identification of in/out-pocket context w.r.t. the best baseline ML models, while for under/over-ground and in/out-door contexts for $\tau^{uo}=313$ m it achieves a balanced accuracy of 4.3% and 1%, respectively,

lower than the best baseline ML model. As our heuristic algorithms consume hardly any memory and have low execution time, they are appropriate for detecting these contexts in the wild.

Our algorithms can be used to annotate online crowd-measurements as they are sensed without any calibration overhead. Context information mined on-board remains private, but can still be used to annotate users' personal trajectories and incentivize them to participate in crowd-measurement campaigns. Our algorithms can also be used offline to contextualize existing datasets and manage their data quality (e.g., detect outlier crowd-measurements).

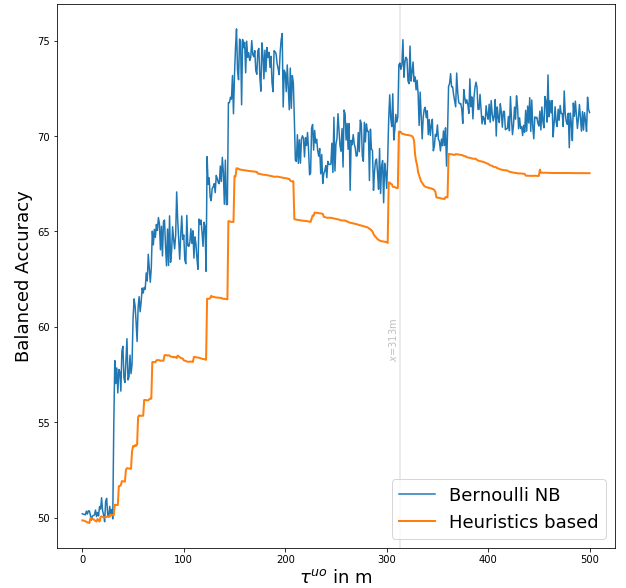


Fig. 1. Affect of τ^{uo} on the balanced accuracy value

As a future study, we would like to integrate our algorithms to the Ambiciti application and possibly exploit additional sensors such as light, that although not reported in the dataset, can be used to improve the accuracy of in/out pocket context inference.

ACKNOWLEDGMENT

We thank Ambiciti for support and for granting us access to the experiment specific data as a part of a research collaboration.

⁶https://github.com/pythonprofilers/memory_profiler

REFERENCES

- [1] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm," *ACM Computing Survey*, vol. 48, pp. 7:1–7:31, August 2015.
- [2] R. Ventura, V. Mallet, V. Issarny, P.-G. Raverdy, and F. Rebhi, "Evaluation and calibration of mobile phones for noise monitoring application," *The Journal of the Acoustical Society of America*, vol. 142, no. 5, pp. 3084–3093, 2017.
- [3] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 1327–1346, November 2017.
- [4] M. Budde, A. Schankin, J. Hoffmann, M. Danz, T. Riedel, and M. Beigl, "Participatory Sensing or Participatory Nonsense?: Mitigating the Effect of Human Error on Data Quality in Citizen Science," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, pp. 39:1–39:23, September 2017.
- [5] J. Freire, A. Bessa, F. Chirigati, H. Vo, and K. Zhao, "Exploring What not to Clean in Urban Data: A Study Using New York City Taxi Trips," *IEEE Data Engineering Bulletin*, vol. 39, no. 2, pp. 63–77, 2016.
- [6] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning Transportation Mode from Raw Gps Data for Geographic Applications on the Web," in *Proceedings of the 17th International Conference on World Wide Web*, WWW'08, (New York, NY, USA), pp. 247–256, ACM, April 2008.
- [7] B. R. Cowan, N. Pantidi, D. Coyle, K. Morrissey, P. Clarke, S. Al-Shehri, D. Earley, and N. Bandeira, "'what can i help you with?': Infrequent users' experiences of intelligent personal assistants," in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '17, (New York, NY, USA), pp. 43:1–43:12, ACM, 2017.
- [8] Y. Vaizman, N. Weibel, and G. Lanckriet, "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, pp. 168:1–168:22, Jan. 2018.
- [9] E. Miluzzo, M. Pap, N. D. Lane, H. Lu, and A. T. Campbell, "Pocket, Bag, Hand, etc.- Automatically Detecting Phone Context through Discovery," in *First International Workshop on Sensing for App Phones (PhoneSense) at SenSys'10*, p. 21–25, 2010.
- [10] K. v. Erum and J. Schöning, "SubwayAPPS: Using smartphone barometers for positioning in underground transportation environments," in *Progress in Location-Based Services 2016*, pp. 69–85, Springer International Publishing, oct 2016.
- [11] S. Ben Mokhtar, A. Boutet, L. Bouzouina, P. Bonnel, O. Brette, L. Brunie, M. Cunche, S. D'Alu, V. Primault, P. Raveneau, H. Rivano, and R. Stanica, "PRIVA'MOV: Analysing Human Mobility through Multi-Sensor Datasets," in *NetMob: Book of Abstracts - Posters*, pp. 19–21, 2017.
- [12] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving Directions Based on Taxi Trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, (New York, NY, USA), pp. 99–108, ACM, 2010.
- [13] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, (New York, NY, USA), pp. 485–492, ACM, 2016.
- [14] J. Yang, E. Munguia-Tapia, and S. Gibbs, "Efficient in-pocket detection with mobile phones," in *Proceedings of ACM conference on Pervasive and ubiquitous computing adjunct publication*, (New York, New York, USA), pp. 31–34, ACM Press, September 2013.
- [15] Y. Kawahara, H. Kurasawa, and H. Morikawa, "Recognizing User Context Using Mobile Handsets with Acceleration Sensors," in *IEEE International Conference on Portable Information Devices*, pp. 1–5, IEEE, May 2007.
- [16] M. Li, P. Zhou, Y. Zheng, Z. Li, and G. Shen, "IODetector: A Generic Service for Indoor/Outdoor Detection," *ACM Transactions on Sensor Networks*, vol. 11, pp. 1–29, December 2014.
- [17] H. Jia, S. Su, W. Knog, H. Luo, and G. Shang, "MobiIO: Push the limit of indoor/outdoor detection through human's mobility traces," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 197–202, IEEE, October 2014.
- [18] M. Okamoto and C. Chen, "Improving GPS-based indoor-outdoor detection with moving direction information from smartphone," in *Adjunct Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp. 257–260, ACM Press, September 2015.
- [19] W. Wang, Q. Chang, Q. Li, Z. Shi, and W. Chen, "Indoor-Outdoor Detection Using a Smart Phone Sensor," *Sensors*, vol. 16, pp. 1563:1–1563:15, September 2016.
- [20] K. Chen and G. Tan, "SatProbe: Low-Energy and Fast Indoor/Outdoor Detection based on Raw GPS Processing," in *IEEE Conference on Computer Communications*, 2017.
- [21] S. Li, Z. Qin, H. Song, C. Si, B. Sun, X. Yang, and R. Zhang, "A lightweight and aggregated system for indoor/outdoor detection using smart devices," *Future Generation Computer Systems*, August 2017.
- [22] Z. Liu, H. Park, Z. Chen, and H. Cho, "An Energy-Efficient and Robust Indoor-Outdoor Detection Method Based on Cell IdentityMap," in *Procedia Computer Science* 56, pp. 189–195, January 2015.
- [23] M. Ali, T. ElBatt, and M. Youssef, "SenseIO: Realistic Ubiquitous Indoor Outdoor Detection System Using Smartphones," in *IEEE Sensors Journal* 18, 9, pp. 3684–3693, May 2018.
- [24] V. Issarny, V. Mallet, K. Nguyen, P. Raverdy, F. Rebhi, and R. Ventura, "Do's and Don'ts in Mobile Phone Sensing Middleware: Learning from a Large-Scale Experiment," in *Proceedings of the 17th International Middleware Conference*, (Trento), pp. 17:1–17:13, December 2016.

APPENDIX

The parameters for Decision Tree classifier for in/out-pocket are *criterion*="gini", *max_depth*=8, *min_samples_leaf*=9, *min_samples_split*=14 while for in/out-door the parameters for Gradient Boosting Classifier are *criterion*='friedman_mse', *init*=None, *learning_rate*=0.5, *loss*='deviance', *max_depth*=5, *max_features*=0.1, *min_sample_leaf*=2, *min_samples_split*=19, *n_estimators*=100, *subsample*=0.15.